

Конспект лекционного занятия от 11.10.2024

Нарушение локальности данных

`#todo` - вставить конспект с лекции 11.10 с ноутбука

Когерентность кэша cache coherency — состояние согласованности данных в кэше и памяти или также в кэшах разных ядер, если они ссылаются на один и тот же участок памяти. Когерентность нарушается операциями записи. Чем быстрее восстанавливается состояние когерентности, тем эффективнее осуществляется обмен данными через разделяемую память (между разными устройствами компьютерной системы, между ядрами процессора).

При работе с параллелизмом возникает такая вещь, как ложное разделение.

Ложное разделение данных false data sharing — **одновременное использование разными ядрами процессора или разными устройствами разных фрагментов одной строки памяти** (например, соседних элементов массива) [1].

Листинг 2.1

```
unsigned T = omp_get_num_threads();
```

Общее количество потоков

Листинг 2.2

```
unsigned s = n / T;
```

Количество элементов массива, которые должен обрабатывать каждый поток. Используется для разбиения массива на приблизительно равные части между потоками.

Листинг 2.3

```
unsigned b = n % T;
```

Остаток от деления общего количества элементов массива `n` на количество потоков `T`. Показывает, сколько потоков получают на один элемент больше базового размера блока `s`.

Использование:

Для первых `b` потоков размер блока увеличивается на 1 (`s + 1`), чтобы справедливо распределить остаток элементов. Для остальных потоков размер блока остается равным `s`.

Пример

пусть $n = 10$, а $T = 3$. Тогда $s = n / T = 3$, а $b = n \% T = 1$

t	0	0	0	0	1	1	1	2	2	2
i	0	1	2	3	4	5	6	7	8	9

Задача `b` — правильно определить начальный индекс участка массива для каждого потока.

Листинг 2.4

```
if (t < b)
    b = ++s * t;
else
    b += s++;
```

Листинг 2.5

```
unsigned t = omp_get_thread_num();
```

`omp_get_thread_num` - функция, которая возвращает номер потока в текущей группе вызывающего потока [4].

Листинг 2.6

```
T = omp_get_num_threads();
```

`omp_get_num_threads` - функция, которая возвращает общее количество потоков в текущей группе [5] (<https://www.openmp.org/spec-html/5.0/openmps111.html>).

Строка компиляции для g++ компилятора листинга 2.1 на intel будет иметь вид:

```
g++ listing\ 2.cpp -o out.exe --std=c++11 --openmp
```

Листинг 2

```
#include <iostream>
#include <cstring>
#include <vector>
#include <iomanip>
#include <omp.h>

unsigned sum_omp_reduce(const unsigned* V, unsigned n, unsigned b, unsigned e) {
    unsigned sum = 0;
    #pragma omp parallel for reduction (+ :sum)
    for (int i = b; i < e; ++i)
        sum += V[i];
    return sum;
}

int main(){
    using std::cin;
    using std::cout;
    using std::endl;
    // Количество элементов в массиве
    unsigned n = 1u << 24;

    // Общее количество потоков
    unsigned T = omp_get_num_threads();
```

```

unsigned s = n / T;
unsigned b = n % T;

#pragma omp parallel
{
    unsigned t = omp_get_thread_num();
    std::vector<unsigned> v(n);

    #pragma omp single
    {
        T = omp_get_num_threads();
        for(int i = 0; i < n; ++i)
            v[i] = i + T;
    }

    if (t < b)
        b = ++s * t;
    else
        b += s++;

    cout << "Thread " << t << "; b is " << b << endl;

    unsigned e = b + s;
    unsigned answer = sum_omp_reduce(v.data(), v.size(), b, e);

    cout << "answer is " << answer << endl;
}

std::cin.get();
return 0;
}

```